Patent Application
Docket No. 92717-00322

CERTIFICATE OF MAILING BY EXPRESS MAIL

"EXPRESS MAIL" Mailing Label No. _EL916518177US
Date of Deposit: ___November 29, 2001_____
I hereby certify that this paper or fee is being deposited with the U.S. Postal Service "Express Mail Post Office to Addressee" service under 37 CFR 1.10 on the date indicated above and is addressed to the Assistant Commissioner for Patents, P. O. Box 2327, Arlington, VA 22202

Type or Print Name: Carol Marstaller

Signature: Carol Marstaller

METHOD AND SYSTEM FOR QUANTITATIVELY ASSESSING PROJECT RISK AND EFFECTIVENESS

RELATED CASES

This application is related to co-pending U.S. patent applications 09/760,339 (Attorney Docket 92717-297), 09/859,320 (Attorney Docket 92717-311) and 09/916,088 (Attorney Docket 92717-314), which are incorporated herein by reference.

BACKGROUND

Field of the Invention

The present invention relates generally to development projects, and, more particularly, but not by way of limitation, to a method and system for quantitatively assessing risk and effectiveness of the development projects.

Background of the Invention

Most business processes can be measured in two ways, efficiency and effectiveness. These two measurements, however,

are difficult to measure quantitatively. Traditionally, these two measurements have been performed in a qualitative manner, which includes a manager generally "feeling" that the business process is operating smoothly and that team members of the business process are working in a cohesive manner. The efficiency issue is addressed in the related applications identified hereinabove, but effectiveness, which is the ability for a service provider to meet a specification of acceptance of a customer, has eluded measurement for quantitative assessment.

During a requirements engagement or development project, a customer or client of a service provider is encouraged to recommend changes and to provide formal comments through use of a formal change proposal process. A change proposal is a request from a client to a service provider, or from the service provider itself, for amending or altering a process or product being developed for a client. The change proposal may be in the form of verbal, paper, or electronic communication. By having client feedback, a direct and continuous indication of the acceptance of the requirements specification is provided. The client feedback also provides a mechanism to assess risk that is introduced to the project when the expectations of the client have not been met, and a change proposal is to be adopted. As understood in the art, change proposals are submitted by review

team members of the client that have responsibility to review and approve the requirements specification deliverables.

Each change proposal submitted for an element or artifact of a requirements engagement or specification has an obvious direct impact in that each change proposal may generate a unit of work by a member of the project team who implements the change to the specified artifact. Even if the change proposal does not generate a task to modify an artifact, at a minimum, a review of the artifact may be necessary. Additionally, each change proposal may have an indirect impact that is not readily obvious as the indirect impact may have a profound effect on project progress. Traditionally, measuring the indirect impact of a change proposal has been performed qualitatively in that the service provider only has been able to provide risk assessment to the client in a general, non-quantifiable manner. While the change proposals are useful in providing feedback for the service provider in terms of (i) risk and (ii) effectiveness, quantitatively assessing the risk and effectiveness for both the service provider and client is not performed as tools specifically designed for such an application previously have been unavailable.

SUMMARY OF THE INVENTION

To overcome the problem of not being able to quantifiably assess (i) risk and (ii) effectiveness of a development project based on change proposals received by a service provider from a client, a method and system for utilizing the change proposals to assess risk and effectiveness have been developed. The risk may include direct and indirect risk created by the change proposals on the project, including the ability of the service provider independently to amend the project based on the request of the change proposal. By quantitatively assessing risks associated with adopting change proposals, the client is able to make a more informed business decision as to whether or not to pursue the changes specified by the change proposals. The effectiveness may be defined as the ability of the service provider to adequately address concerns of the client for the project and may be quantitatively assessed as a function of the change proposals received. The assessment provides the service provider with the ability to objectively and quantitatively monitor how well concerns of the client are being addressed.

One embodiment for quantitatively assessing risk on a project associated with a change proposal by a client of a service provider includes a method and system for assessing risk. The method includes receiving the change proposal of the

client by the service provider. The change proposal may request one or more amendments to be performed on the project being developed by the service provider. One or more elements of the project potentially affected upon the change proposal being approved may be identified based on the amendment(s) of the change proposal. Metric(s) indicative of the potential effects on the project based on the identified element(s) may be generated, where the metric(s) provide an objective risk assessment for the service provider to provide the client.

One embodiment for determining effectiveness of the project development by the service provider for a client includes a method and system for determining the effectiveness. The method includes receiving change proposals from the client by the service provider. The change proposals request amendments to element(s) of the project. A frequency of receipt of the change proposals being received during the course of the project may be monitored. The frequency of the change proposals being received during the course of the project to determine effectiveness of the service provider in the development of the project for the client may be quantitatively evaluated.

Another embodiment for determining effectiveness includes a method and system to determine satisfaction of client expectations of content of the development project by the

service provider. The method may include receiving change proposals from the client by the service provider, where the change proposals request amendments to artifact(s) of the project that are content related. A determination may be made as to the artifact being content oriented. A metric as a function of the proposals being directed to the artifact(s) being content oriented may be directed, where the metric may be indicative of the ability of the service provider to satisfy expectations of the client.

BRIEF DESCRIPTION OF THE DRAWINGS

A more complete understanding of the method and apparatus of the principles of the present invention may be obtained by reference to the following Detailed Description when taken in conjunction with the accompanying Drawings wherein:

FIGURE 1 is an exemplary histogram showing change proposal volume over a project development;

FIGURE 2A is an exemplary artifact tree for a requirement specification;

FIGURE 2B is an exemplary artifact tree at time T_1 for the requirements specification of FIGURE 2A;

FIGURE 3 is an exemplary graph providing a comparison between change proposals in relation to branch stability of the requirements specification of FIGURE 2A;

FIGURE 4 is the exemplary artifact tree of FIGURE 2A at time T_2;

FIGURE 5 is an exemplary graph illustrative of content of the project of FIGURE 2A increasing as the structure of the project stabilizes;

FIGURE 6 is an exemplary graph of potential indirect risk of structure changes late in the development of the project of FIGURE 2A;

FIGURE 7 is an exemplary graph of leaf change proposals for the project of FIGURE 2A during development;

FIGURE 8 is an exemplary bar chart representative of change proposals received during the development of the project of FIGURE 2A;

FIGURE 9 is an exemplary graph of potential impact on the project of FIGURE 2A based on the change proposals received in FIGURE 8;

FIGURE 10 is an exemplary chart indicative of indirect tasks performed due to the change proposals received in FIGURE 8 in relation to potential impact of FIGURE 9;

FIGURE 11 is an exemplary graph indicating direct tasks performed in response to the change proposals received in FIGURE 8 with respect to potential impact of the activities shown in FIGURE 9;

FIGURE 12 is an exemplary graph showing actual impact based on the change proposals received in FIGURE 8 versus potential impact of the activities shown in FIGURE 9;

FIGURE 13 is an exemplary graph providing metrics indicative of indirectly attributable tasks due to change proposals received in FIGURE 8;

FIGURE 14 is an exemplary graph including metrics indicative of directly attributable tasks based on change proposals of FIGURE 8;

FIGURE 15 is an exemplary graph providing metrics based on indirect versus direct tasks;

FIGURE 16 is an exemplary system for maintaining and performing the principles of the present invention as applied to the development project of FIGURE 2A;

FIGURE 17 is an exemplary flow diagram for determining risk assessment according to the principles of the present invention as operated on the exemplary development project of FIGURE 2A;

FIGURE 18 is an exemplary flow diagram describing effectiveness assessment according to the principles of the

present invention as operated on the exemplary development project of FIGURE 2A;

FIGURE 19 is an exemplary block diagram of a class structure for implementing the principles of the present invention;

FIGURE 20A is an exemplary interaction diagram for implementing the principles of the present invention utilizing the class structure of FIGURE 19; and

FIGURES 20B-20J are exemplary interaction diagrams for performing various aspects of the interaction diagram of FIGURE 20A.

DETAILED DESCRIPTION OF THE DRAWINGS

The principles of the present invention will now be described more fully hereinafter with reference to the accompanying drawings, in which embodiments of the principles of the present invention are shown. This invention may, however, be embodied in many different forms and should not be construed as limited to the embodiments set forth herein; rather, these embodiments are provided so that this disclosure will be thorough and complete, and will fully convey the scope of the invention to those skilled in the art.

Service providers, such as consultants and contractors (e.g., software developers), generally utilize a formal change proposal system for receiving feedback from clients to modify a project being developed. The change proposals, however, have traditionally been utilized to provide the service provider with qualitative information as to (i) changes to be made to the project and (ii) effectiveness of the ability of the service provider to satisfy the acceptance criteria of the client. The principles of the present invention provide for utilizing the change proposals in combination with knowledge of the project to quantitatively (i) assess the risk associated with adopting and/or implementing a change proposal and (ii) assess the effectiveness of the service provider in satisfying acceptance criteria of the client.

By quantitatively assessing the risk associated with adopting and/or implementing a change proposal, the consultant and client are able to objectively make an informed business decision as to whether or not to adopt the change proposal. The risk assessment may include identifying elements of the project to be potentially affected upon the change proposal being adopted. A metric indicative of the potential effects on the project based on the identified elements of the project may be generated to provide an objective assessment for the service

provider to provide the client. The metric may be generated using statistical analyses, including using regression analysis to generate correlation coefficients, for example. In the case of the project being a requirements engagement for generating a requirements specification, the identification of elements may include determining descendants of a branch (e.g., determining subsections of a section in a document) requested to be amended by the change proposal. By identifying the elements that have potential to be affected by the change proposal, tasks that are indirectly attributable to the change proposal may be identified and both direct and indirect risk may be assessed before the change proposal is adopted by the client and the service provider.

Effectiveness of a service provider to satisfy the client is also a desirable factor to quantitatively assess from both the service provider and the client's perspectives. The service provider may be interested in such quantitative information so that client relations may be quantitatively monitored and preserved. The client, too, may desire such quantitative information to monitor the progress of the project being developed and to determine whether the service provider is satisfying the requirements of the client. To determine effectiveness, the change proposals being received may be

monitored for frequency of receipt (e.g., the number of change proposals received on a daily or weekly basis) during the course of the project development. The frequency of receipt may be evaluated to quantitatively determine effectiveness of the service provider to satisfy the client. The evaluation may include plotting the frequency or number of change proposals received on a periodic or non-periodic basis on a chart or histogram.

FIGURE 1 is an exemplary histogram or chart 100 for displaying frequency of receipt of change proposals during the course of a project, such as a requirements engagement for developing a requirement specification. Conceptually, during the initial states of the requirements project, the customer review team tends to overcome an initial learning curve that is associated with learning the use of change proposal tools available to the customer review team. Generally, initial customer feedback on a requirements specification is slowly developed. Over time, the content of the specification grows and the customer gains confidence in utilizing the change proposal tools. The customer or client tends to submit change proposals on a more frequent basis as the project progresses. As the requirements specification nears completion, client

feedback is expected to decline as the service provider responds to the change proposals being fed-back.

A theoretical representation of frequency of the change proposals being received by the service provider is shown in the histogram 100. As shown, the rate of frequency of the change proposals increases to a sharp peak 105 and falls off rapidly thereafter. A chi-squared distribution curve as understood in the art may be used to model the change proposal volume provided in the histogram 100. It is assumed that the customer review team has little or no prior experience using a change proposal system. It is further assumed that the requirements specification is released incrementally.

FIGURE 2A is an exemplary requirements artifact tree 200a (inverted tree) including a root artifact 205, branch artifacts 210a-210b (collectively), and leaf artifacts 215a-215d (collectively 215). It should be understood that the leaves 215 are descendants of the root 205 and branches 210, and that the root 205 and branches 210 are ancestors of the leaves 215. The root 205, branches 210, and leaves 215 are individual requirement artifacts of the requirements specification.

FIGURE 2B is an exemplary tree structure 200b of a requirements specification having sections and subsections defined therein. The section 1.0, identified as a root artifact

220, has subsections 1.1 - 1.1.1.5.2.1 identified as branches 225a-225e and leaves 230a-230f. As understood in the art, the section and subsections may be utilized to define a process, system or definition for a development project. Although the tree structure 200b shown is exemplary of a specification or other document, it should be understood that other applications that may be modeled using a tree structure may be utilized in accordance with the principles of the present invention. One such application may include a software development project.

Typically, the structure of a requirements specification, which is embodied by branch requirement artifacts, receives early attention from a requirements development team. For this reason, change proposals are expected to address the specification structure early in the project. As a requirements specification structure (e.g., 200a) reaches stability, modifications of the branch artifacts 210 decline while the age of the branch artifacts 210 increase. FIGURE 3 is an exemplary graph showing a branch stability curve 305 representing change proposals in relation to branch stability. Early in the process, between times S_1 and S_2, branch artifacts 210 have few descendants 215 and modifications to the branch artifacts 210 are expected to have little impact on the project. As structure of the specification develops, the branch stability is shown to

be less stable due to the increase in the number of branches and low age of the branches. As the structure continues to stabilize between the time period between S_2 and S_3, the volume of change proposals that address structure of the tree 200a is expected to decrease. An expected change proposal volume curve 310 represents expected volume for change proposals to be received in relation to the branch stability curve 305. U.S. Patent Serial No. 09/760,339 (Attorney Docket 92717-297), entitled "Method and System for Analyzing and Assessing Progress of a Project" further describes branch artifact stability.

Change proposals may introduce direct and indirect risks as well as potential and actual risks. Reviewers of the requirements specification typically submit change proposals at regular intervals or frequencies during the requirements specification development life cycle. Change proposals are generally directed toward a specific requirements artifact, such as a branch artifact 210. The direct one-to-one relationship between the change proposal and a branch artifact 210, for example, renders assessing the potential direct impact of change proposals as a simplistic exercise. The potential indirect impact of change proposals, however, is more difficult to assess. Further, assessing the actual indirect impact of a change proposal is even more difficult.

As the structure of the requirements specification stabilizes between times S_2 and S_3, the content is expected to grow at a rapid pace as indicated in FIGURE 4, which is the exemplary tree structure 200a at time T_2. The increase in content, which is the total number of leaves 215a-215f, is reflected in the increase in the number of descendants of the branch artifacts 210. As the number of descendants for each branch 210 increases, the potential indirect risk associated with a change proposal directed to the structure of the tree 200b increases in proportion to the age of the artifact for which the change proposal is submitted and the number of descendants that may be impacted by the change proposal. The impact over time on potential indirect risk is represented in FIGURE 5, which is a graph 500 showing that structure stabilizes as the content increases. A stabilization curve 505 is created by dividing the number of leaves 215 by the number of branches 210. The potential indirect impact and actual indirect impact of change proposals may be measured and provided to client reviewers, thereby providing the reviewers with an objective measure of the feedback value being provided in the form of the change proposals. The actual impact of a change proposal is the measure of process effectiveness.

FIGURE 6 is an exemplary graph 600 showing potential indirect risk of structure changes late in the development of the project. As shown, because a single change proposal on a branch artifact 210 may impact a large number of branch 210 and leaf 215 descendants, late structural change proposals introduced the highest potential level of risk and have the potential to become a destabilizing factor for the project as a whole. Often, a late structural change proposal may jeopardize the schedule and budget of the project being developed.

FIGURE 7 is an exemplary graph 700 including the stabilization curve 505 and leaf change proposal curve 705 for the project. As shown, the leaf change proposal curve 705 begins to increase upon leaves 215 being generated by the service provider. As expected, upon the stabilization curve 505 becoming stable at time t_1, the leaf change proposal curve 705 peaks at time t_2 and transitions back to zero toward the end of the project as the content stabilizes. It should be understood that since the leaves, which embody the content, have no descendants, there is little or no potential indirect impact for a change proposal directed toward a leaf artifact 215. Because there is little or no potential indirect impact for a change proposal directed toward a leaf artifact 215, the potential risk associated with content oriented change proposals is directly

proportional to the number of content oriented change proposals that are submitted.

To further understand the potential and actual risk that is introduced to a development project by a change proposal, three items may be assessed: (i) first, the potential impact of the change proposal, where the potential impact includes the number of descendants that may be impacted by the change proposal, (ii) second, the number of actual work elements that have been performed on the artifact specified by the change proposal and the descendants of the artifact, and (iii) third, the correlation between the amount of work actually performed on the artifact specified by the change proposal and the amount of work performed on the descendants of the change proposal.

The principles of the present invention further provide for determining a metric indicative of the ability of the service provider to satisfy the expectation of the client by monitoring change proposals directed toward leafs or content of the development project. Change proposals directed toward content are generally provided after the structure of the development project has stabilized, and indicate a disagreement or dissatisfaction with the content rather than a dissatisfaction with the structure of the development project. Consequently, risk from the content-directed change proposals on the overall

project is low. The metric, however, is valuable in determining client satisfaction with the content. The metric may be generated by utilizing statistical analysis, including using regression analysis and producing correlation coefficients to determine dependency relationships. Further, an indicator may be produced to provide the service provider with an indication of the satisfaction of the client with respect to content being produced by the service provider.

Before further discussing the system and method for performing a risk and effectiveness assessment, a presentation and discussion of results of the system and method are provided. FIGURE 8 is an exemplary histogram 800 depicting the number of change proposals received on any particular date. Frequency of receipt of change proposals may be determined by the number of change proposals received on a date as provided by the bars on the histogram 800. As shown, on 9/19/2001, ten change proposals were received by the service provider, and on 9/21/2001, four change proposals were received by the service provider. Tracking the number of change proposals received allows a service provider to quantitatively assess the impact of a change proposal and qualitatively assess the effectiveness of the service provider in satisfying the expectations of the client.

FIGURE 9 is an exemplary graph 900 indicating the potential impact on the project based on the number of change proposals received in FIGURE 8. On each date, a number of descendants of artifacts toward which the change proposals are directed are counted for that particular date. For example, on 9/21/2001, the number of descendants of the artifacts toward which the four change proposals were directed was four-hundred (i.e., 400).

FIGURE 10 is an exemplary chart 1000 indicative of indirect tasks performed due to the change proposals of FIGURE 8 in relation to potential impact of FIGURE 9. To determine indirect impact, the number of tasks that have been performed on descendants of the artifact toward which the change proposals have been directed are counted. To count the number of tasks performed, the individual units of work performed on a daily basis may be recorded for each artifact. The number of tasks performed on the descendants of the artifacts for which one or more change proposals are submitted may be summed. An indirect task curve 1005 may be plotted against the potential impact curve 905 to show the potential impact versus the indirect tasks performed. As shown, on 9/21/2001, the potential impact for work to be performed was 400, but the actual indirect tasks performed was zero. In this case, the change proposals were rejected by the service provider so that the actual indirect

tasks to be performed for the change proposals submitted were preemptively eliminated. Also shown, on 9/25/2001, the potential impact was roughly 60, but the actual impact was approximately 45. This result shows that at least one change proposal was rejected by either the service provider or the client based on an assessment of the potential impact resulting from the change proposals. It should be understood that the number of indirect tasks are related back to the date of the change proposals that were submitted so that no lag time results in the indirect task curve 1005 relative to the potential impact curve 905.

FIGURE 11 is an exemplary graph 1100 indicating direct tasks performed in response to the change proposals of FIGURE 8 with respect to potential impact of the activities shown in FIGURE 9. As shown, a direct tasks curve 1105 is plotted against the potential impact curve 905. On 9/25/2001, it is shown that the number of direct tasks exceeded the potential impact, which is likely due to multiple direct tasks being performed for one or more change proposals.

FIGURE 12 is an exemplary graph 1200 showing actual impact based on the change proposals of FIGURE 8 versus potential impact of the activities shown in FIGURE 9. As shown, an actual impact curve 1205 is plotted against the potential impact curve

905. The actual impact curve is produced by summing up the indirect and direct tasks 1005 and 1105.

A relationship between actual and potential indirect impact may be established using regression analysis. In order to determine the significance of actual and direct impact as in relation to potential indirect impact, a regression model may be used. The correlation coefficient of the regression equation established by using the number of actual indirect changes as the dependent variable, and the number of potential indirect changes as the independent variable may describe the relationship between potential and actual impact.

A low correlation indicates that the number of artifacts that are actually being modified has a weak relationship to the number of artifacts that have potential to be modified. A weak relationship indicates that volatility of the project is low in that a point of stability may be expected. FIGURE 13 is an exemplary graph 1300 providing a metric indicative of indirectly attributable tasks due to the change proposals of FIGURE 8. In this example, the metric is a correlation coefficient that is generated by correlating the number of indirectly attributable tasks to the number of change proposals. As shown, the correlation coefficient curve 1305 represents correlation coefficients generated from performing a regression analysis.

Before further discussing the results of the principles of the present invention, one embodiment to perform the statistical analysis is discussed. Regression analysis is used to determine dependency between an independent and a dependent variable. For the instant case, independent and dependent variables may be set to provide a desired analysis of risk, stability, or other information of the development project. In performing the regression analysis, statistical equations are used to perform the regression analysis. The regression analysis includes normal regression model equations (equations 1-3) and further includes (i) slope (equation 4) of the regression model equations, (ii) intercept (equation 5) of the regression model equations, (iii) coefficient of determination (equation 6) of the regression equations, and (iv) an equation for the correlation coefficient (equation 7) of the regression equations. The regression analysis is used to compute the regression parameters, develop models of the relationship between desired variables, and assess the strength of the relationships between the desired variables. The equations are expressed as:

$$S_{XX} = \sum X_i^2 - (\sum X_i)^2 / n \qquad (1)$$

$$S_{YY} = \sum Y_i^2 - (\sum Y_i)^2 / n \qquad (2)$$

$$S_{XY} = \sum X Y_i - (\sum X_i)(\sum Y_i)/n \quad (3)$$

$$b_1 = S_{XY} / S_{XX} \quad (4)$$

$$b_0 = \overline{Y} - b_1 \overline{X} \quad (5)$$

$$R^2 = b_1 S_{XY} / S_{YY} \quad (6)$$

$$r = \frac{\sum (X_i - \overline{X})(Y_i - \overline{Y})}{(\sqrt{\sum (X_i - \overline{X})^2} \sqrt{\sum (Y_i - \overline{Y})^2}} . \quad (7)$$

Definitions:

S_{XX} = The sum of the squares of the independent variable values.

S_{YY} = The sum of the squares of the dependent variable values.

S_{XY} = The sum of the products of the independent and dependent variable values.

X = The individual values of the independent variables.

X-bar = The mean of the independent variable values.

Y = The individual values of the dependent variables.

Y-bar = The mean of the dependent variable values.

n = The number of (X,Y) pairs.

b_1 = The slope of the repression equations.

b_0 = The intercept of the regression equations.

r = The sample correlation coefficient.

R^2 = The coefficient of determination, the square of the sample correlation coefficient.

Because the correlation coefficient is relatively low (i.e., below 0±0.2), the project volatility is low, that is, the

relationship between the number of changes performed is weakly attributable to the number of change proposals submitted, and, therefore, risk associated with the change proposals for those particular dates is low. FIGURE 14 is an exemplary graph 1400 including metrics indicative of directly attributable tasks based on the change proposals of FIGURE 8. The correlation coefficients are the metrics and are shown by the correlation coefficient curve 1405. As the correlation coefficients are below ±0.2, the associated risk with the change proposals is low. It should be understood that other metrics, such as R-squared, may be utilized.

In general, a high correlation between actual change impact and potential change impact indicates a strong linear relationship. A strong linear relationship may be translated to say that for every potential change there is a strong likelihood that an actual change occurs, and that extreme volatility in the requirement project exists (i.e., a high risk situation). FIGURE 15 is an exemplary graph 1500 providing metrics on an indirect versus direct task basis. As shown, R-squared is used as the metric and indicates that the correlation is high, which means that the risk on the project for adopting change proposals is high.

FIGURE 16 is an exemplary system 1600 for maintaining and performing the principles of the present invention as applied to the development project of FIGURE 2A. A service provider server 1602a may be coupled to a client server 1602b via a network 1604, such as the Internet. The service provider server 1602a includes a processor 1606 coupled to a memory 1608 and a modem 1610 via a databus 1611. A project datafile 1612 that includes the artifacts of the development project may be stored in a first storage device containing a project summary database 1614. A second storage device may include a risk/effectiveness database 1616, which stores the results of processing for assessing risk and effectiveness of the project and service provider, respectively. It should be understood that the databases 1614 and 1616 may be maintained in single or multiple databases, and be stored on a single storage device. A computing device 1618, such as a personal computer, may be coupled to the service provider server 1602a for performing the operations to generate the project datafile 1612, project summary database 1614, and risk/effectiveness database 1616. The client server 1602b may contain substantially the same hardware as the service provider server 1602a, and communicate with the service provider server 1602a using data packets 1620 as understood in the art.

In operation, the processor 1606 executes at least one software program 1622 that is utilized by a user to generate the project datafile 1612 and databases 1614 and 1616. Alternatively, the project datafile 1612 may be generated utilizing a processor on the computing device 1618 and uploaded on the service provider server 1602a.

FIGURE 17 is an exemplary flow diagram 1700 for determining risk assessment according to the principles of the present invention as operated on the exemplary development project of FIGURE 2A. The process starts at 1705. At step 1710, a change proposal requesting amendments to artifact(s) of the project are received by the service provider from the client. The change proposal may request structure or content of the project. In the case of the project being a requirements specification, the change proposal may request that a branch 210 be moved to a different location or that content, such as a description, be amended, potentially for spelling or grammar, for example. At step 1715, artifact(s) to be potentially affected upon the change proposal being adopted are identified. The artifact(s) may be those directly or indirectly affected by the change proposal. In the case of indirectly affected artifacts, descendants of the directly affected artifacts may be identified. At step 1720, metric(s) indicative of potential

effects on the project are generated to provide objective risk assessment. The metric(s) may be generated using statistical analysis, such as regression analysis. The process ends at step 1725.

FIGURE 18 is an exemplary flow diagram 1800 describing effectiveness assessment according to the principles of the present invention as operated on the exemplary development project of FIGURE 2A. The process starts at 1805. At step 1810, change proposals requesting amendments are received by the service provider from the client. At step 1815, frequency of receipt of change proposals are monitored. The frequency of receipt of change proposals is monitored to determine, for example, how many change proposals are received on a daily or weekly basis. At step 1820, the frequency of receipt of change proposals are evaluated to quantitatively determine effectiveness of the service provider to satisfy the desires of the client. The process ends at step 1825.

FIGURE 19 is an exemplary block diagram of a class structure 1900 for implementing the principles of the present invention. The class structure includes a Task class 1905 containing requirementArtifact 1906 and successors 1907 attributes. The requirementArtifact class 1906 defines particular artifacts that may be identified or modified when

performing a task. The successors attribute 1907 may be a list or container of Task objects that are identified for the particular artifact identified by the requirementArtifact 1906 class. BranchTask 1910, LeafTask, and Change Proposal Task 1920 classes are derived from the Task class 1905 that is used to represent a unit of work. The BranchTask class 1910 defines branch work, the LeafTask class 1905 defines leaf work, and the Change Proposal Task class 1920 defines change proposals submitted by the client. The Change Proposal Task class 1920 further includes directAttributableTasks 1921 and indirectAttributableTasks 1922 attributes that contain the tasks actually performed as a direct or indirect result of a given change proposal.

FIGURE 20A is an exemplary interaction diagram 2000 for implementing the principles of the present invention utilizing the class structure of FIGURE 19 on the processor 1606. The interaction diagram 2000 is a high-level description of particular operations that may be utilized to perform a risk analysis for a particular change proposal. A ProjectAnalyst instance 2004 and Project instance 2006 may be loaded and executed. A user 2002 of the service provider server 1602a may send an AnalyzeClientParticipation method 2008 to the ProjectAnalyst instance 2004. The ProjectAnalyst instance 2004

may determine a change proposal file path at 2010. The ProjectAnalyst instance 2004 may send a message to the Project instance 2006 to analyze the effects of the change proposal.

At 2014, a change proposal file name is selected and client data is set using a method SelectCPFileNameAndSetClientData. Client work knowledge may be generated at 2016. A daily project structure may be built at 2018, where the daily project structure is the structure of the requirements specification in the case of performing a requirements engagement during the project. While the daily project structure may be selectively built to reflect the state of the project on the particular day of the change proposal, a database containing the daily project structures for each day of the project may be maintained, thereby allowing for a simple look-up rather than having to create daily project structures for each analysis.

At 2020, the last tactical datafile that existed is loaded, where the tactical datafile may be a text file containing a listing of all branch and leaf tasks that have been performed, to date. All artifacts, including branch 210 and leaf 215, are assigned or defined as being branches or leaves at 2022. At 2024, potential tasks or work created due to or attributable to the change proposal are identified. An assessment of impact from the change proposal on the project is made at 2026. The

assessment may include counting the tasks that were identified at 2024, for both directly and indirectly attributable tasks. At 2028, regression models for the change proposals may be generated by utilizing results from assessing the impacts of the change proposal as parameters for the regression models. A risk regression analysis is performed at 2030 by utilizing the regression models.

FIGURES 20B-20J are exemplary interaction diagrams for performing various aspects of the interaction diagram of FIGURE 20A.

FIGURE 20B is an exemplary interaction diagram 2000b that further describes the selecting of the change proposal file name and client data setting at 2014 by the Project instance 2006. The SelectCPFileNameAndSetClientData method 2014 may be used to find the most recent change proposal datafile from a pathname argument. The Project instance 2006 creates an input stream and reads the change proposal data. Once the change proposal data has been read, the Project instance 2006 employs a pattern matching capability. Pattern matching capability is the ability to define patterns of knowledge and then construct patterns of data that can be matched to the patterns of knowledge in various ways to accomplish tasks that would otherwise require extensive algorithmic complexity to create a list of change proposal facts

for later use. A client datafile may be read at 2032 to read the change proposal file name and read the client data.

FIGURE 20C is an exemplary interaction diagram 2000c that further describes the building of client work knowledge at 2016 by the Project instance 2006. In the execution of the BuildClientWorkKnowledge method 2016, the Project instance 2006 iterates over the list of change proposal facts or data and matches the fact to a change proposal pattern. The change proposal pattern is a pattern of knowledge that represents change proposal information as it is stored in the change proposal datafile. The change proposal pattern has the following form "("ACCESS" (> dtg) + (> author) (> session) "CHANGE PROPOSAL" "OBJECT" (> oid) (> module) (+ r))". A pattern to be used in creating change proposal tasks." When the Project instance 2006 finds a match of the change proposal facts to the change proposal pattern, a change proposal instance 1920 is initiated to set values for the instance variables (e.g., directAttributableTasks 1921 and indirectAttributableTasks 1922). Once the change proposal instance 2034 has completed setting the instance variables, the change proposal instance 2034 may be added to a hash table in the change proposal attribute 1920 of the Project instance 2006. The Project instance 2006 may build a change proposal task at 2036. At

2038, a pattern match may be executed to determine whether a match of a change proposal task exists. At 2040, a change proposal may be generated by the Project instance 2006 communicating with the change proposal instance 2034. The change proposal may be generated by utilizing the change proposal class 1920. The change proposal instance 2034 may communicate back to the Project instance 2006, and client work information may be added to the change proposal class 1920 at 2044.

FIGURE 20D is an exemplary interaction diagram 2000d that further describes the building of the daily project structure at 2018 by the Project instance 2006. The building of the project structure (e.g., 200b) at 2018 is used to create a replica of the project structure beginning on the first date that a change proposal was submitted and continuing to the current date. The interaction diagram 2000d operates to collect the dates for which there are change proposals, and uses the dates to compose a project datafile name. At 2048, project datafiles are read and the project datafile name may be used to read a particular project datafile at 2050. At 2052, a BuildObjectHierarchy method is utilized to perform a trace on the data read from the project datafile. At 2054, the Project instance 2006 creates an instance 2046 of the RequirementArtifact class 1906. At 2056,

the RequirementArtifact instance 2046 replies to the Project instance 2006, and a pattern match may be performed at 2058 to determine whether a pattern of the data matches the RequireArtifact pattern, the results of the matching process are used to set the attributes of a newly created RequirementArtifact object. At 2060, attributes of the RequirementArtifact class 2046 are communicated to the RequirementArtifact instance 2046. At 2062, RequirementArtifacts instances 2046 are identified to determine whether changes based on the change proposals are to be made. At 2064, the RequirementArtifact instances 2046 are sent an IdentifiedDescendants method for causing the hierarchy of the project artifacts to be constructed. The RequirementArtifacts instances 2046 may be stored in a hash table that is maintained in a projectModules attribute of the Project instance 2006.

FIGURE 20E is an exemplary interaction diagram 2000d that further describes the loading of the last tactical datafile at 2020 by the Project instance 2006. The LoadLastTacticalDatafile method 2066 creates a pathname for the directory in which the tactical data are stored. The pathname is used to create a sorted list of datafiles in the directory by reading the tactical datafiles at 2066. At 2068, work knowledge of the change proposals is created, and work history is built at 2070.

The last file in the directory contains the most recent tactical data, and is used to create instances of the Task class 1905 using the MatchPattern instance 2072. At 2074, a Task instance 2076 is created and a Task class 1905 having attributes may be returned to the Project instance 2006 at 2078. The new Task instances 2076 are placed in a hash table that is maintained in a projectModules of the Project instance 2006.

FIGURE 20F is an exemplary interaction diagram for assigning the artifacts at 2022 by the Project instance 2006. The Project instance 2006 iterates over the daily change proposals, collects the project structure for the date that the change proposals were submitted, and finds the projectRequirementArtifact 1906 for which the change proposal was submitted. At 2080, artifacts are assigned to the change proposals. At 2082, daily candidates of the RequirementArtifacts 1906 are collected. At 2084, the RequirementArtifact attribute of the change proposal instance 2034 is set to the value of the RequirementArtifact 1906 that was found for the change proposal submitted. The process may be repeated for the task instances 2076 at 2086, 2088, and 2090. The assignment of the artifacts process provides for linking the change proposals to the tasks performed.

FIGURE 20G is an exemplary interaction diagram 2000g for attributing tasks to the change proposals at 2092 by the Project instance 2006. Once all the RequirementArtifact instances 2046 have been allocated, the Project instance 2006 causes the change proposal instances 2034 in the change proposal attribute 1920 to find the tasks that are either directly attributable or indirectly attributable to the change proposal. The tasks are recorded in the appropriate change proposal attribute 1920. Change proposals are collected at 2094 and task dates are collected at 2096. A task is directly attributable to a change proposal instance 2034 if the task was performed on the RequirementArtifact 1906 at a point in time after the change proposal 1920 was submitted. At 2098, a DirectlyAttributable method is sent from the Project instance 2006 to the change proposal instance 2034. The change proposal instance 2034 sends the DirectlyAttributable method to the Task instance 2066 at 2100. The Task instance 2066 replies at 2102 a Boolean value to the Change Proposal Instance 2034 to indicate whether the task is or is not directly attributable to the change proposal. An AttributeTask method for the change proposal instance 2034 identifies the task as being directly attributable as defined by the Boolean value.

A task is indirectly attributable to a change proposal if the task was performed at a point in time after the change proposal instance 2034 was submitted and performed on a descendant of the RequirementArtifact 1906 for which the change proposal was submitted. In determining whether the task is indirectly attributable, an InDirectlyAttributable method is communicated from the Project instance 2006 to the Change Proposal Task instance 2034. The IndirectlyAttributable method is communicated from the change proposal instance 2034 to the Task instance 2066 at 2108. A Boolean value may be returned from the Task instance 2066 to the change proposal instance 2034 at 2110 to identify whether the task is IndirectlyAttributable. At 2112, an AttributableTask method identifies whether the task is IndirectlyAttributable based on the Boolean value received by the change proposal instance 2034.

FIGURE 20H is an exemplary interaction diagram 2000h for assessing the change proposal impacts on the project at 2026 by the Project instance 2006. The processes of FIGURES 20B-20J establish the framework of knowledge within which the change proposal impacts may be assessed. At 2014, potential change proposal impact is determined by the Project instance 2006. At 2116, a DeterminePotentialChangeProposalImpact method is communicated from the Project instance 2006 to the change

proposal instance 2034. The change proposal instance 2034 issues a CountDescendants method 2118 to the RequirementArtifact instance 2046. The RequirementArtifact instance 2046 returns a descendant count at 2120 to the change proposal instance 2034, which, in turn, returns the potential impact to the Project instance 2006 at 2122. The Project instance 2006 requests that the change proposal instance 2034 count the number of tasks performed at 2124. The total number of tasks are returned from the change proposal instance 2034 to the Project instance 2006 at 2126. At 2128, the daily potential impact is written or stored and the daily actual impact is written or stored at 2130. It should be understood that the potential impact is determined by counting the total number of descendants of the RequirementArtifact 1906 for which the change proposal was submitted. Change proposals have an actual impact that is represented by the total number of directly and indirectly attributable tasks assigned to the change proposal.

FIGURE 20I is an exemplary interaction diagram 2000i for preparing change proposal regression models at 2028 by the Project instance 2006. At 2134, change proposal regression data is built by the Project instance 2006. Daily regression models are built and populated using the hash table at 2136. At 2138, the Project instance 2006 communicates a

PopulateRegressionParameters method to the RegressionModel instance 2132. At 2138, the regression models are populated with parameters (e.g., independent and dependent variables). The regression models are written to a datafile or repository at 2140. At 2142, the Project instance 2006 communicates to the RegressionModel instance 2132 a WriteToStream method, which implements a modified form of persistence for the RegressionModel class, writing the slope, intercept, correlation coefficient {r} and the coefficient of determination {R-squared} to a text file for later use.

FIGURE 20J is an exemplary interaction diagram 2000j which is used to perform a risk regression analysis at 2030 by the Project instance 2006. Three instances of the RegressionModel class are created for each day of the project duration following the initial date on which the change proposal was submitted, including: (1) the RegressionModel instances provide the capability to model the number of daily indirectly attributable tasks in relation to the number of daily change proposals, (2) the number of daily directly attributable tasks in relation to the number of change proposals, and (3) the number of indirectly attributable tasks in relation to the number of directly attributable tasks, for example. At 2134, change proposal regression model may be built by the Project instance 2006. At

2144, potential and actual risk is compared by the Project instance 2006. At 2146, potential external risk is assessed. Risk factor is determined at 2148 by the Project instance 2006 communicating to the RequirementArtifact instance 2046. At 2150, realized risk is assessed by the Project instance 2006. At 2152, the Project instance 2006 requests the change proposal instance 2034 to determine direct and indirect risk. At 2154, the Project instance 2006 requests the RegressionModel instance 2132 to populate regression parameters. The risk regression models are stored at 2156 by the Project instance 2006.

It should be understood that the exemplary embodiments for implementing the principles of the present invention using the interaction diagrams of FIGURES 20A-20J may be performed using the object oriented structures as shown, but may also be performed by using traditional programming methods as well. In that regard, the principles of the present invention are not dependent upon the particular data structures or classes.

The previous description is of an embodiment for implementing the principles of the present invention, and the scope of the invention should not necessarily be limited by this description. The scope of the present invention is instead defined by the following claims.